Customized Dynamic Host Configuration Protocol

Mr. Sadananda M P and Mr. Sudeep Manohar
Encore Software Private Ltd., Bangalore, India
Email: sadanand119@gmail.com
Jawaharlal Nehru National College of Engineering, Shimoga, India
Email: sudeep_mansh@yahoo.com

***Abstract* - The Dynamic Host Configuration Protocol (DHCP) provides a framework for passing configuration information to hosts on a UDP network. Computers that are connected to IP networks must be configured before they can communicate with other hosts. The most essential information needed is an IP address. DHCP eliminates the manual task by a network administrator. DHCP is based on the Bootstrap Protocol (BOOTP), adding the capability of automatic allocation of reusable network addresses and additional configuration options. DHCP captures the behavior of BOOTP relay agents, and DHCP participants can interoperate with BOOTP participants. Proposed system, i.e., Customized DHCP aims to give the security for DHCP, which was not present in the older one and it uses UDP instead of TCP thus reducing the number of fields as compared to the old DHCP, in turn which decreases the execution time and still providing the basic functionality of the usual DHCP.**

Index Terms – DHCP, BOOTP, DORA, Lease file, security, CUnit, TCP, UDP and IP address.

I. Introduction

DHCP was first defined as a standards track protocol in RFC 1531 in October 1993, as an extension to the Bootstrap Protocol (BOOTP). The motivation for extending BOOTP was that BOOTP required manual intervention to add configuration information for each client, and did not provide a mechanism for reclaiming unused IP addresses.

Dynamic Host Configuration Protocol automates network-parameter assignment to network devices from one or more DHCP servers. Even in small networks, DHCP is useful because it makes it easy to add new machines to the network. When a DHCP configured client (a computer or any other network-aware device) connects to a network, the DHCP client sends a broadcast query requesting necessary information from a DHCP server. The broadcast message is sent because the server address is not known to the client initially.

The DHCP server manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, the name servers, other servers such as time servers, and so forth. On receiving a valid request, the server assigns the computer an IP address, a lease (length of time the allocation is valid), and other IP configuration parameters, such as the subnet mask and the default gateway. The query is typically initiated immediately after booting, and must complete before the client can initiate IP-based communication with other hosts. DHCP is designed to supply DHCP client systems with the configuration parameters that are defined in the Host Requirements RFCs. After obtaining the parameters via DHCP, a DHCP client will be able to exchange packets with any other host in the network.

Not all of these parameters are required for a newly initialized client. A client and server may negotiate for the transmission of only those parameters required by the client or specific to a particular subnet. DHCP allows but does not require the configuration of client parameters not directly related to the IP protocol. DHCP also does not register newly configured clients with the Domain Name System (DNS).

II. The DORA process

DORA stands for Discover, Offer, Request and Acknowledge. DORA is the process which is used by DHCP for allotting the IP address. It is a four way communication between the Client and Server. The DORA concept is shown in the fig. 1.

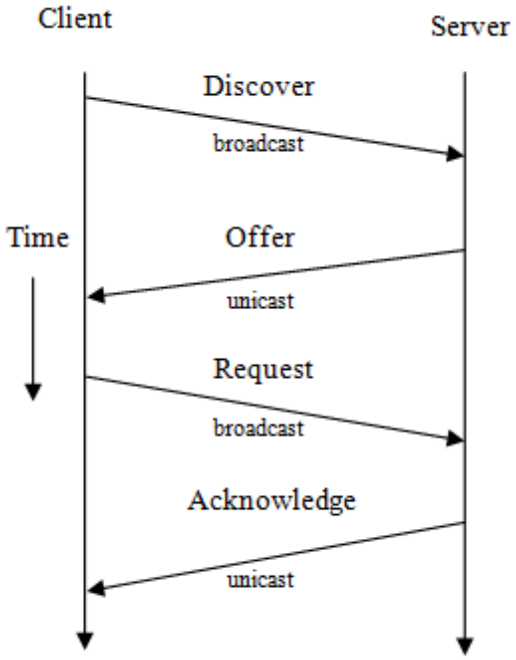

Figure 1. DORA concept

The steps involved in the DORA concept are as below

a) Client makes a UDP Broadcast to the server with a DHCPDiscover, or Discover packet.

b) DHCP offers to the client. The server sends a DHCP Offer including the IP address and other configuration parameters (DHCP Options).

c) In response to the offer Client requests the server. The client replies DHCPRequest, unicast to the server, requesting

DOI: 01.IJCOM.02.03. 548

the offered address.
d) The server sends DHCPAck acknowledging the request which is the clients final permission to take the address as offered. Before sending the ack the server checks once again where the offered address is still available and that the parameters match the clients request. After allotting the IP the server marks the address taken.

III. Customized DHCP

In the Customized DHCP, the four way communication is reduced to two way communication. Instead of the DORA concept, a new concept is being used which has only two steps, request and response. The code length is reduced by removing the unnecessary fields in the packet which increases the speed of execution thus decreasing the time taken to allot the IP address.

Here UDP is used instead of TCP and thus decreasing the number of fields in the packet. The Customized DHCP provides security in which, the protocol can detect any unauthorized client trying to connect to the network and can keep the user away from trusted network.
This security feature implemented in Customized DHCP is not available in the usual DHCP and it is an added feature in the Customized DHCP. For the security feature implementation, a secret code is used in the client, which should match with the server's secret code.

IV. Implementation

A. Project Functional Overview

The Customized DHCP assigns IP address based on the range assigned by admin. The range may vary according to admin. Within the range the server should generate random IP address and assign to the client and it also provides static IP address based on client interest. For the static IP address the client should specify the MAC address and server hostname with same IP address.

B. IP Address Allocation Methods

Depending on the implementation, the DHCP server may have three methods of allocating IP-addresses:
• Dynamic allocation: A network admin assigns a range of IP addresses to DHCP, and each client computer on the LAN is configured to request an IP address from the DHCP server during network initialization. The request-and-grant process uses a lease concept with a controllable time period, allowing the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed.
• Automatic allocation: The DHCP server permanently assigns a free IP address to a requesting client from the range defined by the administrator. This is like dynamic allocation, but the DHCP server keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.
• Static allocation: The DHCP server allocates an IP address based on a table with MAC address/IP address pairs, which are manually filled in by the network administrator. Only requesting clients with a MAC address listed in this table will be allocated an IP address.
Any of the above methods can be used, but in Customized DHCP implementation, dynamic allocation method is used to provide quick allocation of IP addresses to the clients..

C. Protocols Used

1) Internet Protocol: The Internet Protocol (IP) is the principal communications protocol used for relaying datagram (packets) across an inter-network using the Internet Protocol Suite. IP is the primary protocol in the Internet Layer of the Internet Protocol Suite and has the task of delivering datagram from the source host to the destination host solely based on their addresses. For this purpose, IP defines addressing methods and structures for datagram encapsulation. Fig. 2 shows the structure of a IP packet.
2) User Datagram Protocol: The User Datagram Protocol (*'*UDP) is one of the core members of the Internet Protocol Suite. With UDP, computer applications can send messages, in this case referred to as datagram, to other hosts on an Internet Protocol (IP) network without requiring prior communications to set up special transmission channels or data paths. UDP applications use datagram sockets to establish host-to-host communications. An application binds a socket to its endpoint of data transmission, which is a combination of an IP address and a service port. Fig. 3 shows the UDP packet structure.

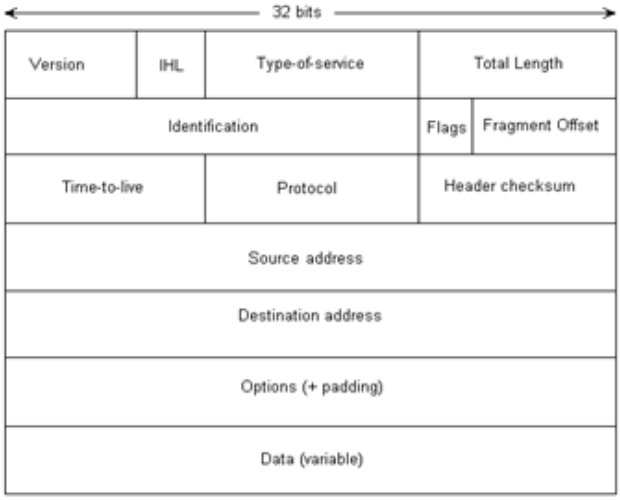

Figure 2. Internet Protocol

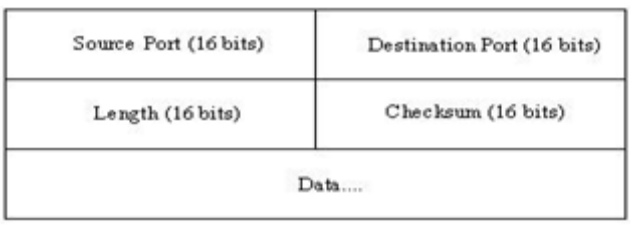

Figure 3. User Datagram Protocol

3) Bootstrap Protocol: In computer networking, the Bootstrap Protocol, or BOOTP, is a network protocol used by a network client to obtain an IP address from a configuration server. The BOOTP protocol was originally defined in RFC 951. BOOTP is usually used during the bootstrap process

DOI: 01.IJCOM.02.03. 548

when a computer is starting up. A BOOTP configuration server assigns an IP address to each client from a pool of addresses. BOOTP uses the User Datagram Protocol (UDP) on IPv4 networks only. Fig. 4 shows the BOOTP packet structure.

4) Ethernet frame: A data packet on an Ethernet link is called an Ethernet frame. A frame begins with Preamble and Start Frame Delimiter. Following which, each Ethernet frame continues with an Ethernet header featuring destination and source MAC addresses. The middle section of the frame is payload data including any headers for other protocols (e.g. Internet Protocol) carried in the frame. The frame ends with a 32-bit cyclic redundancy check which is used to detect any corruption of data in transit. Fig. 5 shows Ethernet Frame structure.

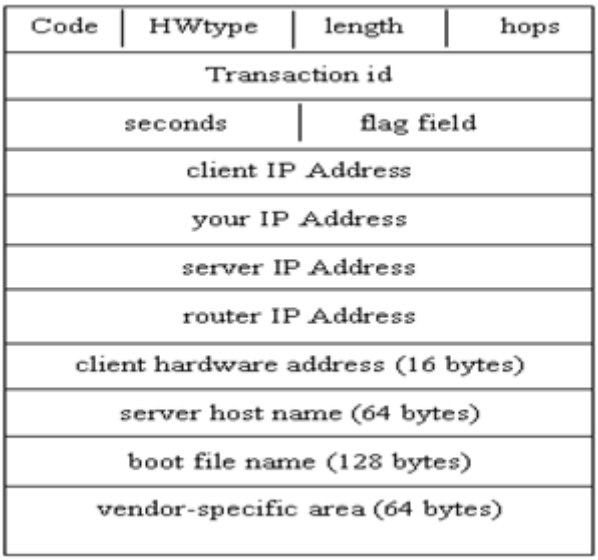

Figure 4. Bootstrap Protocol

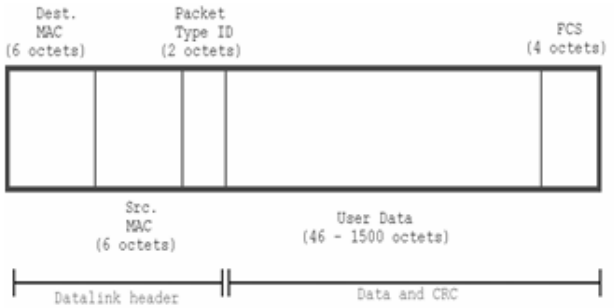

Figure 5. Ethernet frame

D. How It Works

- Initially client and server communicate each other by creating sockets.
- Server system has to retrieve client's mandatory configuration detail (MAC address).
- Client has to retrieve necessary information and fill the same in Ethernet frame, IP, UDP and BOOTP.
- For the transmission of datagram to the remote destination it uses broadcast with parsing data mechanism.
- The server will receive the request packet from client then binds address with the port address.
- Server receives all the information send by client and check for available IP and assigns dynamically to the client.

E. Workflow Diagram

The work flow diagram shown in Fig. 6 describes the full functionality of the project. In Customized DHCP two way communications takes place. Client will broadcast the request with all basic information of the client system. The server will receive the request and reads all the client information. It validates the secret code and checks whether the client is authorized user or not. Finally the server unicasts the response to the intended client system with the IP address and other configuration settings.

F. Customized DHCP Configuration Structure File

```
Host abc
{
        Hardware Ethernet xx:xx:xx:xx:xx:xx
        Fixed address 192.168.5.20;
        Optioned Hostname abc;
}

Subnet 192.168.5.0 netmask 255.255.255.0
{
        Range 192.168.5.3 to 192.168.5.254;
        Optional subnet mask 255.255.255.0;
        Optional broadcast address 192.168.5.254;
        Optional routes 192.168.5.1;
        Optional domain name server 10:20:30:40;
        }
```

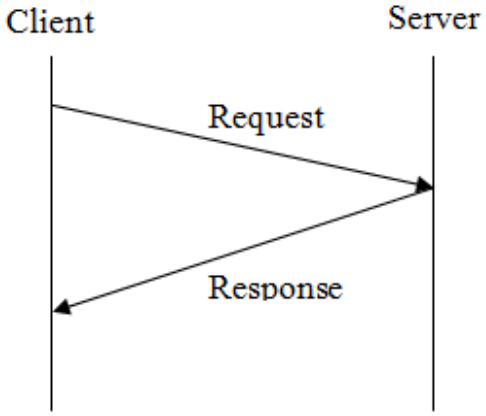

Figure 6. Work Flow Diagram

By refereeing above structure there are two parts.

Part 1 (Host abc)

- Used for static IP address.
- Uses Static allocation to allot the IP addresses.

Part 2 (Subnet)

- A network administrator assigns a range of IP addresses to DHCP.
- The DHCP server permanently assigns a dynamic IP address to a requesting client from the range.

DOI: 01.IJCOM.02.03.548

G. Providing security for DHCP server.

```
Host abc
{
        Hardware Ethernet xx:xx:xx:xx:xx:xx;
        Fixed address 192.168.5.20;
        Optioned Hostname abc;
        Server Hostname xyz;
}

Subnet 192.168.5.0 netmask 255.255.255.0
{
        Range 192.168.5.3 to 192.168.5.254;
        Optional subnet mask 255.255.255.0; Optional
        broadcast address 192.168.5.254;
        Optional routes 192.168.5.1;
        Optional domain name server 10:20:30:40;
        Server Hostnames xyz;
}
```

DHCP security structure includes an extra field, the server hostname as the secret code. When client broadcasts the request with all the basic information, the client has to send the specified secret code of the server. The server will receive the request and compares the host name. If the server hostname matches, then server will assign the first range of IP address. If it does not match, then the server will assign second range of IP address. First ranges have the authority to access all the data from the server or the organization, but the second range will require some access permission. The diagram is Fig. 7 shows the two levels of security given to users.

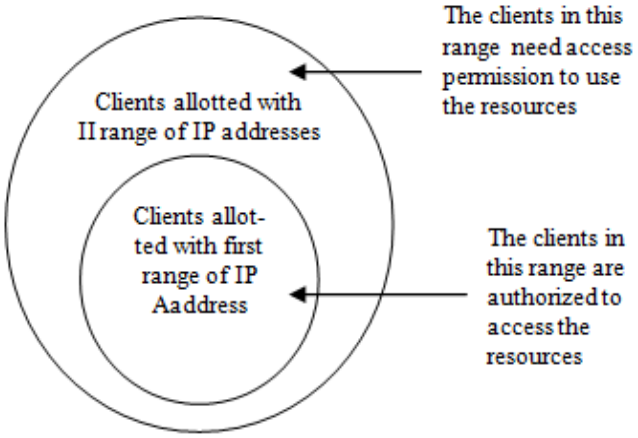

Figure 7. Customized DHCP security architecture

Any other client who is unauthorized will not be knowing the server hostname, which is kept secret within the network perimeter. If such a client tries to request the server, it can be easily identified, since the server hostname will not be there in the request part. Thus the request it is denied from giving the access permissions, but still, it is allotted with an IP address. Thus, the implementation provides two category of IP address allocation which provides security for the clients which are connected using the first range of IP addresses. The implementation depicts the function of a firewall. Hence the Customized DHCP acts as a firewall which identifies the unauthorized clients.

H. Leased File

DHCP servers update their databases frequently. It is very hard to maintain the consistency among these databases. When the server system reboots, since the hard disk is not in the working condition, the database can not be used to store the information. So the LEASED FILE is used in cache memory to store the information of IP addresses. It holds file structure which contains the IP addresses, MAC addresses and leased time of the each individual system. This file structure is stored in the server's cache memory. Whenever admin reboots the system the information stored in the file structure will be automatically erased. Before allotting any IP address for a client which has made a request, the server first check the list of IP address allotment, which has already been made, and then decides which IP address to be allotted to the client. So, using file structure to store the data has overcome the problem present in using a database.

V. TESTING

A. Introduction

Testing is an important phase in the development life cycle of the product. This is the phase where all the types of errors are detected. Errors that are found and corrected are recorded for future references. Thus, a series of testing is performed on the system before it is ready to use.

B. CUnit Test

CUnit is a lightweight system for writing, running and administering unit tests in C. It provides C programmers a basic testing functionality with a flexible variety of user interfaces. CUnit is built as a static library which is linked with the user's testing code. It uses a simple framework for building test structures, and provides a rich set of assertions for testing common data types. In addition, several different interfaces are provided for running tests and reporting results.

B. Test Cases

The implementation is tested using CUnit test module. It provides various test cases on which the implementation is tested upon and in all the test positive results have been obtained. Table. I shows the list of various test cases that are considered while testing.

DOI: 01.IJCOM.02.03.548

TABLE I. TEST CASES

Test cases	result
Test of parse_Hardware_Ethernet ( )	Passed
Test of parse_Fixed_Address ( )	Passed
Test of parse_hostname ( )	Passed
Test of parse_serverhostname ( )	Passed
Test of parse_range ( )	Passed
Test of parse_Optional_mask ( )	Passed
Test of parse_Optional_broadcasr ( )	Passed
Test of parse_optroutes ( )	Passed
Test of parse_optdns ( )	Passed
Test of dhcp_clent ( )	Passed
Test of dhcp_lease_file ( )	Passed
Test of dhcp_server_hostname_verify ( )	Passed
Test of dhcp_broadcast ( )	Passed
Test of dhcp_send_pkt ( )	Passed
Test of dhcp_receive_pkt ( )	Passed
Test of dhcp_assignip ( )	Passed
Test of dhcp_testing_interface ( )	Passed
Test of dhcp_server ( )	Passed
Test of dhcp_server_receive_pkt ( )	Passed
Test of dhcp_server_unicast_reply ( )	Passed
Test of dhcp_parse_pkt ( )	Passed
Test of dhcp_fill_bootp ( )	Passed
Test of dhcp_send_pkt ( )	Passed

C. Run Summary

The overall testing run summary of the tests conducted is shown in the Table. II. In all the test types the implementation found to be positive without any negative results.

TABLE II. RUN SUMMARY

Type	Total	Ran	Passed	Failed	Inactive
Suites	1	1	n/a	0	0
Tests	23	23	23	0	0
Asserts	46	46	46	0	n/a

VI. FUTURE ENHANCEMENT

Though the Customized DHCP is added with some new features, it still lacks some features that are yet to be implemented.

- The project is sufficiently working in command prompt. But it would be more user friendly if it runs in a GUI environment.
- Security can be upgraded to level three, i.e., a third range of IP addresses can be given to one more category of clients which increases the ease of authorization.

VII. CONCLUSION

The implementation of Customized Dynamic Host Configuration Protocol provides security feature and also reduces the number of communication messages between the client and the server. Also by using UDP most of the unnecessary fields are removed, thereby enhancing the efficiency in terms of execution time taken. The customized DHCP provides high level of security by authorizing the hostname sent by the client to the server. Thus the customized DHCP works like a firewall or a Intrusion Detection System which detection intrusions and provides security. Hence the customized DHCP provides an efficient way for allotting IP addresses to the client systems by reducing the communication between the client and server and by increasing the level of security.

REFERENCES

[1] Behrouz A Forouzan, *"Data Communications and Networking"*, 4th ed., New York: Tata McGraw Hill, 2006, pp. 618-620.
[2] Nader F Mir, *"Computer and Communication Networks"*, 3rd ed., Pearson Education, 2009, pp. 174.
[3] William Stallings, *"Data and Computer Communications"*, 8th ed., New Jersey: Prentice Hall, pp. 34 – 40.
[4] Behrouz A Forouzan, *"TCP/IP Protocol Suite"*, 2nd ed., New York: Tata McGraw Hill, 2003, pp. 481 – 490.
[5] W. Richard Stevens, *"UNIX Network Programming"*, 2nd ed., vol 1.
[6] Brian W Kernighan and Dennis M Ritchie, *"The C Programming Language"*, 2nd ed., Prentice Hall, 1988.

DOI: 01.IJCOM.02.03. 548